Joachim Rang, Rainer Niekamp

A component framework for the parallel solution of the incompressible Navier-Stokes equations with Radau-IIA methods

Informatikbericht Nr. 2014-07

Institute of Scientific Computing
Carl-Friedrich-Gauss-Fakultät
Technische Universität Braunschweig

Braunschweig, Germany

This document was created Juli 2014 using LaTeX 2_ε.

Institute of Scientific Computing
Technische Universität Braunschweig
Hans-Sommer-Straße 65
D-38106 Braunschweig, Germany

url: www.wire.tu-bs.de
mail: wire@tu-bs.de

A component framework for the parallel solution of the incompressible Navier-Stokes equations with Radau-IIA methods

Joachim Rang, Rainer Niekamp

Institute of Scientific Computing, TU Braunschweig
j.rang@tu-bs.de

Abstract

An efficient solution strategy for the simulation of incompressible fluids needs adequate and accurate space and time discretisation schemes. In this paper for the space discretisation we use an inf–sup stable finite element method and for the time discretisation Radau-IIA methods of higher order, which have the advantage that the pressure component has convergence order s in time, where s is the number of internal stages. The disadvantage of this approach is that we have a high computational amount of work, since large nonlinear systems of equations have to solved. In this paper we use a transformation of the coefficient matrix and the simplified Newton method. This approach has the effect that our large nonlinear systems split into smaller ones, which can now also be solved in parallel. For the parallelisation of the code we use the software component technology and the Component Template Library (CTL). Numerical examples show that high order in the pressure component can be achieved and that the proposed solution technique is very effective.

1 Introduction

Many physical or engineering problems can be described with partial differential equations, such as the simulation of time–dependent laminar flows, which can be described by Navier–Stokes equations. The accurate and fast solution of these equations is the core of many numerical simulations of complex processes in nature and industry. The discretisation in space and time is needed for the computation of the solution. In the case of the incompressible Navier–Stokes equations often an inf–sup stable finite element method is used [14]. It was demonstrated in a number of numerical studies, e.g. in [33, 18, 20], that the pair of second order velocity Q_2 and first order discontinuous pressure P_1^{disc} on quadrilateral and hexahedral meshes is among the best performing finite element methods. Thus, the Q_2/P_1^{disc} finite element is a popular choice if finite element methods are used in the simulation of incompressible flows [15].

Many simulations of incompressible flows use explicit or simple implicit time-stepping schemes, such as the backward Euler scheme, the Crank–Nicolson scheme or the fractional–step θ–scheme. We will concentrate in this paper on implicit schemes which are appropriate for laminar flow simulations and which avoid the nasty Courant–Friedrichs–Lewy (CFL) condition. One possibility to solve such problems are one-step methods [16, 35] such as implicit Runge–Kutta methods, since a high order of convergency can be achieved (see [31]). Often an order reduction phenomenon can be observed if stiff problems or DAEs are solved (see [23] and [16]). Moreover the costs for the linear algebra are very high. Therefore often diagonally implicit Runge-Kutta (DIRK) methods or Rosenbrock–Wanner (ROW) methods are used for the simulation of fluids. But in this case the order reduction in the pressure is much stronger. Many papers are studying the order reduction phenomena and derive further order conditions to reduce this effect. One possibility is the consideration of the Prothero–Robinson example. In [30] and [28] better ROW methods are developed and numerical studies show that full order can be reached for the Prothero–Robinson example. Similar results for SDIRK and ESDIRK methods are presented in [30] and [29].

In this paper we want to use very accurate methods, i.e. methods which have a high order of convergence. Therefore fully implicit Runge–Kutta methods may be a good choice. But these methods need a high computational effort, since in every timestep a nonlinear system of dimension ns has to be solved, where n is the dimension of the problem and s is the number of internal stages of the Runge–Kutta method. In the last decades several papers have discussed the efficient solution of the nonlinear or linear equations. Here we use a transformation from Butcher [7] and Bickart [2] of the coefficient

matrix of the Runge–Kutta method. If a simplified Newton method is applied this splitting leads to s complex valued systems of dimension n. An application of this technique for Radau methods can be found in [17] and [31]. These nonlinear systems can be solved directly with the help of LU-decompositions and back- and forward substitutions. But here we have to store s complex LU-decompositions. Therefore we use a parallel framework where each node solves one of the s linear systems.

The implementation is based on the Component Template Library (CTL) which supports the easy developement of distributed as well as parallel applications. This is a C++ template library using template meta-programming to hide as many as possible technical details from the programmer. It has been successfully applied in computational applications such as multi-physics simulation [4, 3, 25, 34, 32], multi-scale simulation [26, 24], stochastic finite element analysis [21, 10] and optimisation [22]. The framework described in this paper was parallelised with only small changes to an already existing serial code by defining a component interface for the linear solver and by outsourcing this time expensive part to parallel working software components.

In this paper we solve the incompressible Navier–Stokes equations with the help of high order Radau-IIA methods. For the solution of the nonlinear systems we use the simplified Newton method with a transformation of the coefficient matrix. This approach gives us the possibility to solve the s systems in parallel and therefore we have a very effective method, which is shown with the help of different numerical examples.

The paper is structured as follows: First we give a short introduction into the discretisation of the incompressible Navier-Stokes equations. The Finite Element method for the space discretisation and Runge–Kutta methods for the time discretisation are presented. The nonlinear system is formulated and solution techniques are presented. Finally we give some examples which show that Radau-IIA methods have a high accuracy in the pressure component and work much more efficient than 3rd and 4th order DIRK and ROW methods.

2 The discretisation of the incompressible Navier-Stokes equations

2.1 Space discretisation

Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, be a bounded domain and $T > 0$. The motion of an incompressible flow is modeled by the incompressible Navier–Stokes

equations, which are given in dimensionless form by

$$\begin{aligned} \mathbf{u}_t - Re^{-1}\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in } (0,T]\times\Omega \\ \nabla\cdot\mathbf{u} &= 0 \quad \text{in } [0,T]\times\Omega, \end{aligned} \tag{1}$$

where $\mathbf{u}$ denotes the velocity, p the pressure, $\mathbf{f}$ represents body forces and parameter Re is the Reynolds number. The system of equations (1) has to be closed with appropriate initial and boundary conditions. If Dirichlet conditions are prescribed on the whole boundary $\partial\Omega$, a condition for the pressure, such as $\int_\Omega p(\mathbf{x})\, d\mathbf{x} = 0$, has to be added.

For simplicity of presentation, we consider the case that (1) is equipped with homogeneous Dirichlet boundary conditions in $[0,T]$. Then, the velocity ansatz and test spaces can be chosen the same in the weak formulation of (1) as well as in the finite element method. Let $V = (H_0^1(\Omega))^d$, $Q = L_0^2(\Omega)$, then the time–continuous weak or variational problem reads as: Find $(\mathbf{u}, p) \in V \times Q$ such that

$$\begin{aligned} (\mathbf{u}_t, \mathbf{v}) + (Re^{-1}\nabla\mathbf{u}, \nabla\mathbf{v}) + ((\mathbf{u}\cdot\nabla)\mathbf{u}, \mathbf{v}) - (p, \nabla\cdot\mathbf{v}) &= (\mathbf{f}, \mathbf{v}) \quad \forall\, \mathbf{v}\in V \\ (\nabla\cdot\mathbf{u}, q) &= 0 \quad \forall\, q \in Q, \end{aligned} \tag{2}$$

where $(\cdot,\cdot)$ denotes the inner product in $(L^2(\Omega))^d$, $d \in \{1,2,3\}$. Finite element methods are a standard approach to perform the spatial discretization of (2), [15]. The unique solvability of the arising discrete system requires that the velocity finite element space V_h is sufficiently large compared to the pressure finite element space Q_h. The precise description of this property is the so–called inf–sup condition [14]

$$\inf_{q_h\in Q_h} \sup_{\mathbf{v}_h\in V_h} \frac{(q_h, \nabla\cdot\mathbf{v}_h)}{\|q_h\|_{L^2}\|\nabla\mathbf{v}_h\|_{L^2}} \geq \beta > 0. \tag{3}$$

To avoid technical difficulties in the presentation of the methods, we will consider conforming finite element spaces, i.e. $V^h \subset V$ and $Q^h \subset Q$. The space–discretized Navier–Stokes equations read as follows: Find $(\mathbf{u}, p) \in V \times Q$ such that

$$\begin{aligned} &(\mathbf{u}_{h,t}, \mathbf{v}_h) + (Re^{-1}\nabla\mathbf{u}_h, \nabla\mathbf{v}_h) \\ &+((\mathbf{u}_h\cdot\nabla)\mathbf{u}_h, \mathbf{v}_h) - (p_h, \nabla\cdot\mathbf{v}_h) = (\mathbf{f}_h, \mathbf{v}_h) \quad \forall\, \mathbf{v}_h \in V_h \\ &(\nabla\cdot\mathbf{u}_h, q_h) = 0 \quad \forall\, q_h \in Q_h. \end{aligned} \tag{4}$$

System (4) can be represented in algebraic form. Let $\{\boldsymbol{\phi}_{i,h}\}_{i=1}^{dN_u}$ be a basis of V_h and $\{\psi_{i,h}\}_{i=1}^{N_p}$ be a basis of Q_h, where N_u is the number of degrees of freedom for each component of the velocity and N_p is the number of degrees

of freedom for the pressure. Then, the solution of (4) can be written in the form

$$\mathbf{u}_h(t,\mathbf{x}) = \sum_{i=1}^{dN_u} u_{i,h}(t)\boldsymbol{\phi}_{i,h}(\mathbf{x}), \quad p_h(t,\mathbf{x}) = \sum_{i=1}^{N_p} p_{i,h}(t)\psi_{i,h}(\mathbf{x}).$$

For shortness, all algebraic forms will be given for the two dimensional case. The extension to three dimensions is straightforward. Defining the following matrices and vectors

$$\begin{array}{rcll} (M)_{ij} & = & (\varphi_j,\varphi_i), & i,j=1,\ldots,N_u, \\ (A(\mathbf{u}_h))_{ij} & = & Re^{-1}(\nabla\varphi_j,\nabla\varphi_i) & \\ & & +(\mathbf{u}_h^{(1)}\partial_x\varphi_j + \mathbf{u}_h^{(2)}\partial_y\varphi_j,\varphi_i), & i,j=1,\ldots,N_u, \\ (B_1)_{ij} & = & -(\partial_x\varphi_i,\psi_j), & i=1,\ldots,N_u, j=1,\ldots,N_p, \\ (B_2)_{ij} & = & -(\partial_y\varphi_i,\psi_j), & i=1,\ldots,N_u, j=1,\ldots,N_p, \\ (f_k)_i & = & (f_k,\varphi_i), & i=1,\ldots,N_u, \quad k=1,2, \end{array} \tag{5}$$

the algebraic representation of (4) is

$$\begin{pmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\mathbf{u}}_h^{(1)} \\ \dot{\mathbf{u}}_h^{(2)} \\ \dot{\mathbf{p}}_h \end{pmatrix} = \begin{pmatrix} \mathbf{f}_h^{(1)} \\ \mathbf{f}_h^{(2)} \\ 0 \end{pmatrix} - \begin{pmatrix} A(\mathbf{u}_h) & 0 & B_1 \\ 0 & A(\mathbf{u}_h) & B_2 \\ B_1^T & B_2^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_h^{(1)} \\ \mathbf{u}_h^{(2)} \\ \mathbf{p}_h \end{pmatrix}.$$

Matrix M is called mass matrix and A stiffness matrix. Let t_{m+1} denote the new discrete time, t_m the previous discrete time and $\tau_m = t_{m+1} - t_m$, $m = 0, 1, 2, \ldots$ the length of the time step. For simplicity of notation we omit in the following the index h. Moreover we denote by $\mathbf{u}_m = (\mathbf{u}_m^{(1)}, \mathbf{u}_m^{(2)})^\top$ the numerical approximation of the solution $\mathbf{u}(t_m)$.

2.2 Time discretisation

Application to ODEs. We start our considerations with the implicit initial value problem

$$M\dot{\mathbf{u}} = \mathbf{f}(t,\mathbf{u}), \quad \mathbf{u}(t_0) = \mathbf{u}_0, \tag{6}$$

where M is a regular matrix. A Runge-Kutta (RK) method for the implicit

ODE (6) is given by

$$M\mathbf{k}_i = \mathbf{f}\left(t_m + c_i\tau, \mathbf{u}_m + \tau_m \sum_{j=1}^{s} a_{ij}\mathbf{k}_j\right), \quad i = 1, \ldots, s, \tag{7}$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \tau_m \sum_{i=1}^{s} b_i \mathbf{k}_i, \tag{8}$$

where τ is a given timestep size, s is the number of internal stages and a_{ij}, b_i, and c_i are the coefficients of the RK-method, which should be determined in such a way that the method has a sufficiently high order convergency [8, 16, 35]. The order of the RK-method can be determined with the so-called simplifying conditions from Butcher [6], which are defined as follows.

Definition 2.1. *(see [6]). An s-stage RK-method satisfies the simplifying conditions if the conditions*

$$\begin{aligned}
B(p): &\quad \sum_{i=1}^{s} b_i c_i^{k-1} &&= 1/k, && k = 1, \ldots, p,\\
C(q): &\quad \sum_{j=1}^{s} a_{ij} c_j^{k-1} &&= c_i^k/k, && i = 1, \ldots, s, k = 1, \ldots, q,\\
D(r): &\quad \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} &&= b_j(1 - c_j^k)/k, && j = 1, \ldots, s, k = 1, \ldots, r
\end{aligned}$$

are fulfilled.

The condition $B(p)$ is equivalent to a quadrature rule with nodes c_i and weights b_i, which integrates polynomials of degree $p-1$ exactly. The conditions $C(q)$ have the following meaning: The intermediate values $\mathbf{k}_i$ are integrated exactly by a quadrature rule with weights a_{ij} and nodes c_i, which integrates polynomials of degree q exactly.

Theorem 2.2. *(see [8, 35]) An RK-method with s internal stages has the convergence order p if the simplifying conditions $B(p)$, $C(l)$, and $D(m)$ with*

$$p \leq \min\{l + m + 1, 2l + 2\}$$

are satisfied.

For the proof we refer to the book of Butcher [8]. Next we derive the coefficients of Radau-IIA methods, which need the roots of the shifted Legendre

polynomial of degree s, i.e.

$$P_s(2t-1) = \frac{1}{s!}\frac{d^s}{dt^s}[t^s(t-1)^s].$$

With respect to the $L^2(0,1)$-scalar product the polynomial $P_s(2t-1)$ is orthogonal to all polynomials of degree $< s$. The roots of the Legrendre polynomials P_s can be found in the book of Abramowitz and Stegun [1] or can be computed with a computer algebra tool. It can be proven that the roots are pairwise distinct. From this fact it follows that the Vandermonde matrix

$$V_s = (V_{ij}) := (c_i^{j-1}) = \begin{pmatrix} 1 & c_1 & c_1^2 & \dots & c_1^{s-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & c_s & c_s^2 & \dots & c_s^{s-1} \end{pmatrix}, \qquad i,j = 1,\dots,s$$

is regular. In the case of the Radau methods we need the following theorem.

Theorem 2.3. *Let be given a Runge–Kutta method with $p = 2s-1$. Then the nodes c_i of the RK method are given by the roots of the polynomial*

$$P_{s,\xi}(2x-1) = P_s(2x-1) + \xi P_{s-1}(2x-1), \quad \xi \in \mathbb{R}.$$

Proof. see [36]. □

Here we are intested in the case $\xi = -1$, which lead to the Radau-IIA methods with $c_s = 1$ (see [13]). The condition $B(p)$ reads as $\mathbf{b}^\top \mathbf{c}^k = 1/k$, where the vector $\mathbf{c}^k$ is defined as $\mathbf{c}^k = (c_1^k, \dots, c_s^k)^\top$. The condition $C(q)$ can be written as $A\mathbf{c}^{k-1} = \mathbf{c}^k/k$, where $A = (a_{ij})_{i,j=1}^s$. The nodes b_i are then uniquely determined by the conditions $B(1)$, ..., $B(s)$, i. e. by

$$\mathbf{b}^\top \mathbf{e} = 1, \mathbf{b}^\top \mathbf{c} = 1/2, \dots, \mathbf{b}^\top \mathbf{c}^{s-1} = 1/s.$$

This system can be written in matrix-vector notation as

$$\mathbf{b}^\top V_s = \mathbf{e}_H^\top := \left(1, \frac{1}{2}, \dots, \frac{1}{s}\right).$$

Multiplying from the right with the inverse of V_s generates our nodes b_i, i.e. $\mathbf{b}^\top = \mathbf{e}_H^\top V_s^{-1}$. Next we determine the matrix A with the help of conditions $C(1), \dots, C(s)$, which can be written as

$$A\mathbf{e} = \mathbf{c}, A\mathbf{c} = \mathbf{c^2}/2, \dots, A\mathbf{c}^{s-1} = \mathbf{c^s}/s,$$

or in matrix notation by $AV_s = C$, where

$$C := (c_{ij}) = \frac{1}{j} c_i^j, \qquad i, j = 1, \ldots, s.$$

As it is shown in [31] the method can be equipped with an embedded method. Therefore we set

$$\tilde{\mathbf{e}}_H^\top := \left(1, \frac{1}{2}, \ldots, \frac{1}{s-1}, 0\right).$$

Then the nodes $\tilde{b}_i$ are given simply by

$$\tilde{\mathbf{b}}^\top = \tilde{\mathbf{e}}_H^\top V_s^{-1}$$

and the embedded method is of order $s-1$. Finally the Butcher table is given by

$\mathbf{c}$	CV_s^{-1}
	$\mathbf{e}_H^\top V_s^{-1}$
	$\tilde{\mathbf{e}}_H^\top V_s^{-1}$

and the Radau-IIA methods with 2 and 3 internal stages are given by

$\frac{1}{3}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{3}{4}$	$\frac{1}{4}$
	$\frac{3}{4}$	$\frac{1}{4}$
	1	0

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2+3\sqrt{6}}{225}$
1	$\frac{16-6\sqrt{6}}{36}$	$\frac{16+6\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{16-6\sqrt{6}}{36}$	$\frac{16+6\sqrt{6}}{36}$	$\frac{1}{9}$
	-1	$1-\frac{7}{12}\sqrt{6}$	$1+\frac{7}{12}\sqrt{6}$

Application to DAEs of index 2. Consider the DAE

$$M\dot{\mathbf{u}} = \mathbf{F}(t, \mathbf{u}, \mathbf{p}) \tag{9}$$

$$0 = \mathbf{G}(t, \mathbf{u}). \tag{10}$$

We will assume in the following that the matrix $\partial_{\mathbf{u}}\mathbf{G} M^{-1} \partial_{\mathbf{p}}\mathbf{F}$ is non–singular, where $\partial_{\mathbf{u}}\mathbf{G}$ denotes the Jacobian of $\mathbf{G}$ with respect to the space variable $\mathbf{u}$, and so on. In this case the DAE (9)–(10) has the differentiation index 2 (see [5, 16]). For index-2 DAEs only an initial condition for $\mathbf{u}_0$ is given. However, the application of an IRK–method to (9)–(10) requires also the definition of an initial value for $\mathbf{p}_0$. To this end, the algebraic constraint (10) is differentiated, which leads to

$$0 = \mathbf{G}_t(t, \mathbf{u}) + \mathbf{G}_u(t, \mathbf{u})\dot{\mathbf{u}}.$$

Inserting this result into (9) yields

$$-\mathbf{G}_t(t,\mathbf{u}) = \mathbf{G}_u(t,\mathbf{u})\dot{\mathbf{u}} = \mathbf{G}_u M^{-1}\mathbf{F}(t,\mathbf{u},\mathbf{p}). \tag{11}$$

To derive an RK–method for the DAE (9)–(10) instead of the algebraic constraint (10) one considers the differential equation

$$\varepsilon I\dot{\mathbf{p}} = \mathbf{G}(t,\mathbf{u}), \quad \varepsilon > 0,$$

For the system of this equations together with (9) an RK–method can be applied. By letting $\varepsilon \to 0$, the RK–method for the DAE (9)–(10) is obtained [16]. It reads as follows:

$$M\mathbf{k}_i = \mathbf{F}(t_m + c_i\tau_m, \mathbf{U}_i, \mathbf{P}_i), \quad \mathbf{U}_i = \mathbf{u}_m + \tau_m \sum_{j=1}^{s} a_{ij}\mathbf{k}_j, \tag{12}$$

$$0 = \mathbf{G}(t_m + c_i\tau_m, \mathbf{U}_i), \quad \mathbf{P}_i = \mathbf{p}_m + \tau_m \sum_{j=1}^{s} a_{ij}\mathbf{l}_j, \quad i = 1,\ldots,s, \tag{13}$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \sum_{i=1}^{s} b_i\mathbf{k}_i, \quad \mathbf{p}_{m+1} = \mathbf{p}_m + \sum_{i=1}^{s} b_i\mathbf{l}_i, . \tag{14}$$

In our case the coefficient matrix A of the Butcher–table is regular and the values $\mathbf{l}_j$ in (12)–(14) are well–defined. For convergence results we refer to the book of Hairer and Wanner [16], where it is proven that Radau-IIA methods converge with order $2s-1$ for the velocity component and with order s for the pressure component.

Application to the Navier–Stokes equations. In the case of the incompressible Navier–Stokes equations our RK method reads as

$$\begin{pmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{k}_i^{(1)} \\ \mathbf{k}_i^{(2)} \\ \mathbf{l}_i \end{pmatrix} = \begin{pmatrix} \mathbf{f}_h^{(1)}(t_m + c_i\tau) \\ \mathbf{f}_h^{(2)}(t_m + c_i\tau) \\ 0 \end{pmatrix} - \begin{pmatrix} A(\mathbf{U}_i^{(1)}) & 0 & B_1 \\ 0 & A(\mathbf{U}_i^{(2)}) & B_2 \\ B_1^T & B_2^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}_i^{(1)} \\ \mathbf{U}_i^{(2)} \\ \mathbf{P}_i \end{pmatrix},$$

where

$$\mathbf{U}_i^{(1)} := \mathbf{u}_m^{(1)} + \tau \sum_{j=1}^{s} a_{ij} \mathbf{k}_j^{(1)},$$

$$\mathbf{U}_i^{(2)} := \mathbf{u}_m^{(1)} + \tau \sum_{j=1}^{s} a_{ij} \mathbf{k}_j^{(2)},$$

$$\mathbf{P}_i^{(1)} := \mathbf{p}_m + \tau \sum_{j=1}^{s} a_{ij} \mathbf{l}_j.$$

The initial pressure $\mathbf{p}_0$ can be computed from (11).

2.3 Solution of the nonlinear systems

Application to ODEs If we apply an implicit Runge–Kutta method to a system of ODEs, then in every timestep a large non-linear system of equations has to be solved. We start our considerations with the implicit Runge–Kutta method (7)–(8), which can be written as

$$\begin{pmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_s \end{pmatrix} = \begin{pmatrix} \mathbf{f}\left(t_m + c_1\tau, \mathbf{u}_m + \tau \sum_{j=1}^{s} a_{1j}\mathbf{k}_j\right) \\ \vdots \\ \mathbf{f}\left(t_m + c_s\tau, \mathbf{u}_m + \tau \sum_{j=1}^{s} a_{sj}\mathbf{k}_j\right) \end{pmatrix}. \tag{15}$$

Equation (15) forms a coupled non-linear system of equations with dimension $ns \times ns$, which can be solved with the simplified Newton method. By $\mathbf{f}_u$ we denote the Jacobian of $\mathbf{f}$ at the point $(t_m, \mathbf{u}_m)$, and then the simplified Newton method reads as

$$\begin{pmatrix} I - \tau a_{11}\mathbf{f}_u & \dots & -\tau a_{1s}\mathbf{f}_u \\ \vdots & & \vdots \\ -\tau a_{s1}\mathbf{f}_u & \dots & I - \tau a_{ss}\mathbf{f}_u \end{pmatrix} \begin{pmatrix} \Delta\mathbf{k}_1^{(\nu+1)} \\ \vdots \\ \Delta\mathbf{k}_s^{(\nu+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{f}\left(t_m + c_1\tau, \mathbf{u}_m + \tau \sum_{j=1}^{s} a_{1j}\mathbf{k}_j^{(\nu)}\right) \\ \vdots \\ \mathbf{f}\left(t_m + c_s\tau, \mathbf{u}_m + \tau \sum_{j=1}^{s} a_{sj}\mathbf{k}_j^{(\nu)}\right) \end{pmatrix} - \begin{pmatrix} \mathbf{k}_1^{(\nu)} \\ \vdots \\ \mathbf{k}_s^{(\nu)} \end{pmatrix} \tag{16}$$

with $\Delta\mathbf{k}_i^{(\nu+1)} = \mathbf{k}_i^{(\nu+1)} - \mathbf{k}_i^{(\nu)}$, $i = 1, \ldots, s$. If we introduce the Kronecker symbol $A \otimes B$ (see [35]) defined by

$$A \otimes B = \begin{pmatrix} a_{11}B & \ldots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \ldots & a_{mn}B \end{pmatrix},$$

where A and B are matrices, we can write our linear systems as

$$(I_s \otimes I_n - \tau A \otimes \mathbf{f}_u)\Delta\mathbf{K}^{(\nu+1)} = \mathbf{F}^{(\nu)} \tag{17}$$

with $\Delta\mathbf{K}^{(\nu+1)} := (\Delta\mathbf{k}_1^{(\nu+1)}, \ldots, \Delta\mathbf{k}_s^{(\nu+1)})^\top$ and

$$\mathbf{F}^{(\nu)} := \begin{pmatrix} \mathbf{f}\left(t_m + c_1\tau, \mathbf{u}_m + \tau \sum_{j=1}^s a_{1j}\mathbf{k}_j^{(\nu)}\right) \\ \vdots \\ \mathbf{f}\left(t_m + c_s\tau, \mathbf{u}_m + \tau \sum_{j=1}^s a_{sj}\mathbf{k}_j^{(\nu)}\right) \end{pmatrix} - \begin{pmatrix} \mathbf{k}_1^{(\nu)} \\ \vdots \\ \mathbf{k}_s^{(\nu)} \end{pmatrix}.$$

The linear system (17) can be solved with only one LU decomposition and then in each Newton iteration a forward and a backward substitutions can be applied, since the coefficient matrix does not change during the timestep.

As starting values for the Newton method often the setting $\mathbf{k}_i^{(0)} := 0$ is used. This is of course not the best choice. In [16] the starting values are computed with the help of interpolation. Therefore let

$$\mathbf{z}_i := \mathbf{U}_i - \mathbf{u}_m = \tau \sum_{j=1}^s a_{ij}\mathbf{k}_j.$$

Then it holds

$$\mathbf{z}_i = \mathbf{u}(t_m + c_i\tau_m) - \mathbf{u}_m + \mathcal{O}(\tau^{\eta+1}),$$

if the simplifying condition $C(\eta)$ is satisfied for some $\eta \leq s$. In the case of the Radau-IIA methods we have $c_1 \neq 0$, and we can consider the interpolation polynomial of degree s, defined by

$$\mathbf{q}(0) = 0, \mathbf{q}(c_i) = \mathbf{z}_i, \quad i = 1, \ldots, s,$$

since $c_i = \sum_{j=1}^s a_{ij}$ holds (see simplifying condition $C(1)$).

2.4 The transformation of the coefficient matrix

Since the solutions of the huge nonlinear systems are very expensive, we try to reduce these costs by splitting these systems into s smaller ones. One possibility is a transformation of coefficient matrix A, which was derived independently from Bickart [2] and Butcher [7]. We multiply system (17) from the left with $P \otimes I$ and from the right with $Q \otimes I$. The matrices P and Q are choosen in such a way that the product

$$(P \otimes I)(I_s \otimes I_n - \tau A \otimes \mathbf{f}_u)(Q \otimes I)$$

is a lower block triangular matrix. Let us assume that the coefficient matrix A is regular, which is true for the Gauß-Legendre, the Radau-IA, the Radau-IIA and the Lobatto-IIIC methods. Then the eigenvalues of the coefficient matrix A are non-zero and it is possible to compute the Jordan canonical of A, which is given by

$$T^{-1}A^{-1}T = \begin{pmatrix} \lambda_1^{-1} & & 0 & \\ \mu_1 & \lambda_2^{-1} & & \\ & \ddots & \ddots & \\ 0 & & \mu_{s-1} & \lambda_s^{-1} \end{pmatrix},$$

where $\mu_i = 0$, if $\lambda_i \neq \lambda_{i+1}$ and any other real number. Next we introduce the diagonal matrix D given by $D := \mathrm{diag}(\lambda_1, \ldots, \lambda_s)$ and select $P := DT^{-1}A^{-1}$, $\quad Q := T$. Then we have

$$PQ = \begin{pmatrix} 1 & & 0 & \\ \epsilon_1 & 1 & & \\ & \ddots & \ddots & \\ 0 & & \epsilon_{s-1} & 1 \end{pmatrix}, \qquad \epsilon_i \in \{0, 1\}.$$

Note that the matrices D, P and Q are complex valued matrices in our cases. If we apply this transformation to our application of the incompressible Navier-Stokes equations we get the complex valued iteration matrix

$$I_s \otimes M - \tau L \otimes C,$$

where L is a lower triangular matrix with complex entries.

3 Numerical results

For the computations a linux cluster core with a 3 GHz system processor was used. The linear systems are solved directly with an LU-decompostion. The simulations are performed with the code MooNMD [19], with the linear solver package UMFPACK [12, 11] and with the CTL. We compare our results with some diagonally implicit Runge–Kutta (DIRK) and Rosenbrock–Wanner (ROW) methods. The DIRK methods satisfy further order conditions for index-2 DAEs to obtain better convergence results for stiff ODEs [37]. Morever ESDIRKPR53, ESDIRKPR63, and ESDIRKPR74 are constructed in such a way that they converge with full order for the stiff Prothero–Robinson example [29]. In Table 1 we collect the properties of the selected methods.

Table 1: Properties of the selected ESDIRK methods

Name	s	p	$\lvert R(\infty)\rvert$	$\lvert\tilde{R}(\infty)\rvert$	reference
ESDIRK32	4	2	0.33	1	[37]
ESDIRKPR53	5	3	0	0	[29]
ESDIRKPR63	6	3	0	0	[29]
ESDIRKPR74	6	4	0	0	[29]
ROS34PRW	4	3	0	0.25	[27]
RODASPR	6	4	0	0	[28]

3.1 An example where only a discretization error in time occurs

In our first example of the incompressible Navier–Stokes equations the right-hand side f, the initial condition u_0 and the non-homogeneous Dirichlet boundary conditions are chosen such that

$$\begin{aligned} u_1(t,x,y) &= \sin(t)(y^2+x), \\ u_2(t,x,y) &= \sin(t)(x^2-y), \\ p(t,x,y) &= \exp(-t)(x+y-1) \end{aligned}$$

is the solution of (1). Moreover we set $Re = 1$, $\Omega = (0,1)^2$ and solve the problem in the time interval $(0, 1/10]$. We use the Q_2/P_1^{disc} discretisation on a uniform mesh which consists of squares with an edge length $h = 1/512$. Note

that for any t the solution can be represented exactly by discrete functions. Hence, all occurring errors will result from the temporal discretisation. During the calculations we have to deal with 132,098 d.o.f. for the velocity and 49,152 d.o.f. for the pressure. As time steps we use $\tau = \frac{16}{10 \cdot 2^k}$, $k = 0, \ldots, 6$. The numerical results are presented in Figure 1. We plot the timestep size τ

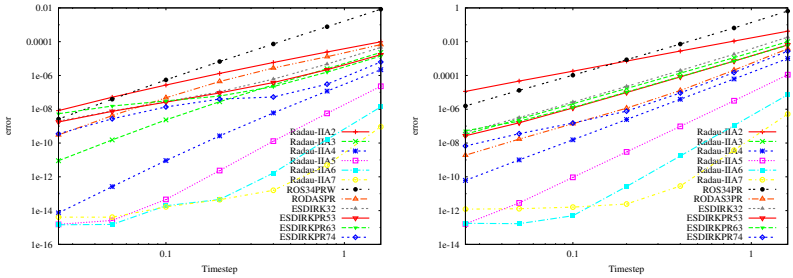

Figure 1: τ versus error for (1) velocity u (left) and pressure p (right)

against the velocity and the pressure error. It can be observed that the higher order Radau-IIA methods compute the numerical solution more effectively than the DIRK and ROW methods. The results for the numerical order of convergence for the pressure component are presented in Table 2. The Radau-IIA methods have the advantage that the convergence order for the pressure component equals to the stage order s. Therefore we get a much faster convergence than for the DIRK and ROW methods. We can observe, too, that the ESDIRKPR53, ESDIRKPR63 and the ESDIRKPR74 methods converge with order 3 and 4, resp., since they meet further order conditions [31]. A similar observation can be made for ROS34PR and RODASPR since these methods satisfy further order conditions (see [30] and [31]).

3.2 A vortex decay problem

The following problem can be found in [9] and has the solution

$$
\begin{aligned}
u_1 &= -\cos(n\pi x)\sin(n\pi y)\exp(-2n^2\pi^2 t/\hat{\tau}),\\
u_2 &= \sin(n\pi x)\cos(n\pi y)\exp(-2n^2\pi^2 t/\hat{\tau}),\\
p &= -\frac{1}{4}(\cos(2n\pi x) + \cos(2n\pi y))\exp(-4n^2\pi^2 t/\hat{\tau}).
\end{aligned}
$$

For the relaxation time $\hat{\tau} = Re = 1000$, this is a solution of the Navier-Stokes equations (1) consisting of an array of opposite signed vortices which decay exponentially as $t \to \infty$.

Table 2: Numerical order of convergency for the pressure component

	τ	$\frac{8}{5}$	$\frac{4}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{10}$
Radau-IIA2	$\|\underline{\epsilon}\|$	4.18e-02	1.09e-02	2.80e-03	7.10e-04	1.79e-04
	q_{num}		1.94	1.96	1.98	1.99
Radau-IIA3	$\|\underline{\epsilon}\|$	1.00e-02	1.14e-03	1.37e-04	1.70e-05	2.11e-06
	q_{num}		3.14	3.05	3.02	3.01
Radau-IIA4	$\|\underline{\epsilon}\|$	1.01e-03	6.20e-05	3.91e-06	2.46e-07	1.55e-08
	q_{num}		4.02	3.99	3.99	3.99
Radau-IIA5	$\|\underline{\epsilon}\|$	1.11e-04	3.18e-06	9.66e-08	2.99e-09	9.33e-11
	q_{num}		5.13	5.04	5.01	5.00
Radau-IIA6	$\|\underline{\epsilon}\|$	7.22e-06	1.09e-07	1.71e-09	2.69e-11	4.96e-13
	q_{num}		6.05	6.00	5.99	5.76
Radau-IIA7	$\|\underline{\epsilon}\|$	5.20e-07	3.72e-09	2.83e-11	2.45e-12	1.61e-12
	q_{num}		7.13	7.04	3.53	0.61
ROS34PRW	$\|\underline{\epsilon}\|$	6.50e-01	6.48e-02	7.23e-03	8.53e-04	1.03e-04
	q_{num}		3.33	3.16	3.08	3.04
RODASPR	$\|\underline{\epsilon}\|$	3.78e-03	2.07e-04	1.29e-05	1.16e-06	1.39e-07
	q_{num}		4.19	4.01	3.47	3.06
ESDIRK32	$\|\underline{\epsilon}\|$	1.85e-02	1.81e-03	1.96e-04	2.25e-05	2.65e-06
	q_{num}		3.35	3.21	3.12	3.08
ESDIRKPR53	$\|\underline{\epsilon}\|$	6.42e-03	7.06e-04	8.22e-05	9.77e-06	1.19e-06
	q_{num}		3.18	3.10	3.07	3.04
ESDIRKPR63	$\|\underline{\epsilon}\|$	6.58e-03	7.26e-04	8.62e-05	1.06e-05	1.34e-06
	q_{num}		3.18	3.07	3.03	2.98
ESDIRKPR74	$\|\underline{\epsilon}\|$	7.10e-03	4.12e-04	2.53e-05	1.58e-06	1.02e-07
	q_{num}		4.11	4.03	4.00	3.95

In the numerical tests we have used Dirichlet boundary conditions on the whole boundary. The right hand side f, the initial condition u_0 and the non-homogeneous Dirichlet boundary conditions are chosen such that

$(u_1, u_2, p)^\top$ is the closed form solution of (1) for a given set of parameters. We present computations for the relaxation time $\hat{\tau} = 1$, the vortex configuration $n = 4$, the final time $\bar{t} = 1$ with the Reynolds number 1 on a fixed spatial grid. The grid consits of squares with edge length $h = 1/512$. On this grid, the Q_2/P_1^{disc} finite element discretization possesses $132,098$ d.o.f. for the velocity and $49,152$ d.o.f. for the pressure. We carried out computations

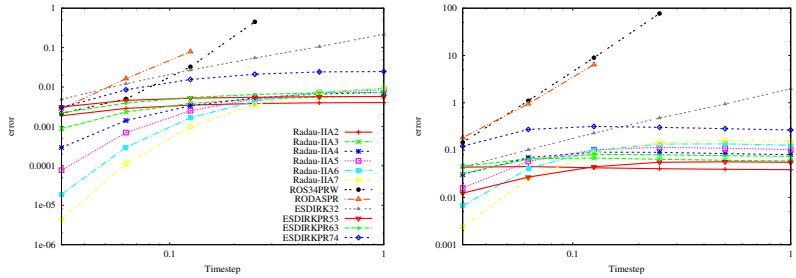

Figure 2: τ versus error for (1) velocity u (left) and pressure p (right)

with the equidistant time steps $\tau_N = \frac{1}{N}$ with $N \in \{0, \ldots, 5\}$. The results are presented in Figure 2, where we plot the timestep size τ against the error of the velocity and the pressure error. The ROW methods do not work for large stepsizes such as $\tau = 1$ and $\tau = 1/2$. Therefore these results are omitted in the graphics. For small timestep sizes it can be observed that the higher order Radau-IIA methods produce much better results than the lower order methods. Moreover the computational time is smaller, too.

3.3 Flow around a cylinder

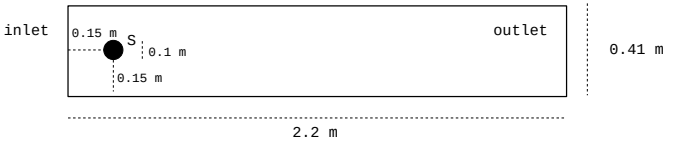

Figure 3: Flow around a cylinder, the channel with the cylinder

The flow around a cylinder which will be considered was defined as a benchmark problem in [33] and studied numerically in detail in [18]. Figure 3 presents the flow domain. The right hand side of the Navier-Stokes equations (1) is $f = 0$, the final time is $\bar{t} = 8$, and the inflow and outflow boundary

conditions are given by

$$u(t,0,y) = u(t,2.2,y) = 0.41^{-2}\sin(\pi t/8)(6y(0.41-y),0)\; m\; s^{-1},\; 0 \le y \le 0.41.$$

On all other boundaries, the no-slip condition $u = 0$ is prescribed. The Reynolds number of the flow, based on the mean inflow, the diameter of the cylinder and the prescribed viscosity $\nu = 10^{-3}\, m^2\, s^{-1}$ is $0 \le Re(t) \le 100$.

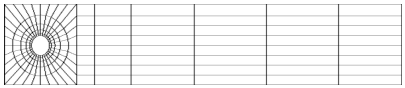

Figure 4: Flow around a cylinder, the coarsest grid (level 0)

The coarsest grid (level 0) is presented in Figure 4. All computations are carried out on level 4 of the spatial grid refinement resulting in 107,712 velocity d.o.f. and 39,936 pressure d.o.f.

In this paper we take as benchmark value the difference of the pressure between the front and the back at the cylinder at the final time $p(8, 0.15, 0.2) - p(8, 0.25, 0.2)$ (see [33]). Reference values for this difference on a fine grid are given in [18, 20]. We apply all methods with an adaptive timestep control.

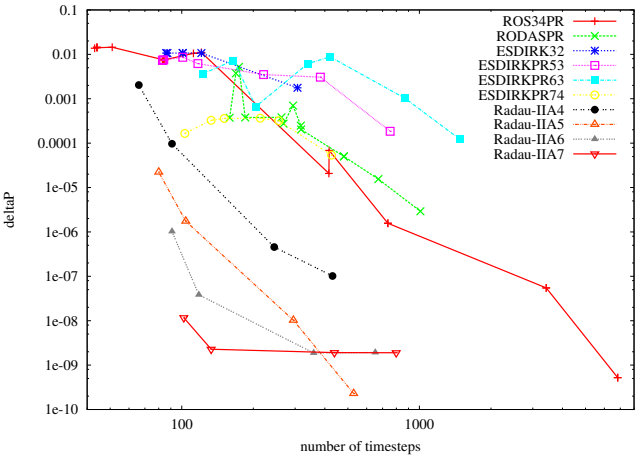

Figure 5: number of timesteps versus error for (1): pressure difference

For the computation for the next timestep we use the PI-Controller. In the case of the Radau-IIA methods we use embedded formulas which are

introduced in [31]. In Figure 5 we present the numerical results and plot the number of timesteps against the numerical error, because it is difficult to compare the computing time of these methods. In the case of the Radau-IIA methods the linear systems are solved in parallel, i. e. we use s cores, where s is the number of internal stages, and in the case of DIRK and ROW methods no parallel approach is used. It can again be observed that the higher order Radau-IIA methods need less timesteps than our DIRK and ROW methods to reach the same accuracy. In our experience the computing time for one timestep is longer in the case of the Radau-IIA methods, but we need less of them. Hence, for the computations in this paper the Radau-IIA methods are much more effective than the DIRK and ROW methods. This effect can be observed in Figure 6, too. In this visualisation we have plotted the time t against the timestep size τ for three different simulations, which provide approximately the same accuracy. It can observed that the Radau-IIA methods are using larger timestep sizes.

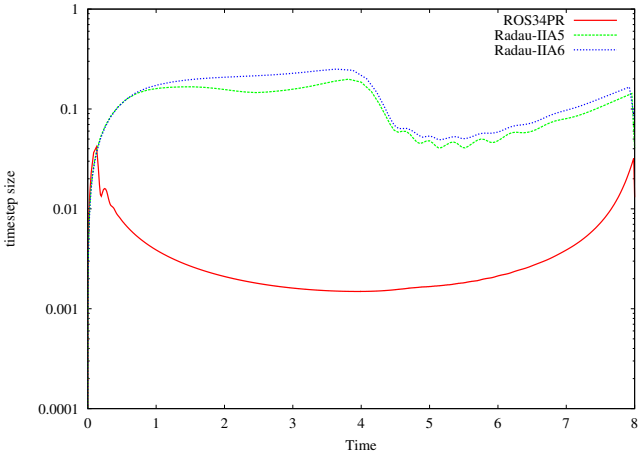

Figure 6: time t versus timestep length τ

4 Summary and Outlook

In this paper we solve the incompressible Navier–Stokes equations with Radau-IIA methods. These are fully implicit Runge–Kutta methods which need the solution of huge nonlinear systems of equations. Therefore we use a

simplified Newton method for the solution of these systems and apply a matrix transformation such that huge dimensional linear systems split up into s smaller ones. We use the Component Template Library (CTL) to solve these s linear systems with LU decompositions in parallel. Our numerical examples such as the well-known Benchmark examle from Turek and Schäfer show that for appropriate order of the Radau-IIA methods we get an effective solution strategy.

References

[1] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions.* John Wiley & Sons, 1984.

[2] T. A. Bickart. An efficient solution process for implicit Runge-Kutta methods. *SIAM J. Numer. Anal.*, 14:1022–1027, 1977. doi:10.1137/0714069.

[3] P. Birken, K. Quint, S. Hartmann, and A. Meister. Choosing norms in adaptive FSI calculations. *PAMM*, 10(1):555–556, 2010. Available from: http://dx.doi.org/10.1002/pamm.201010270, doi:10.1002/pamm.201010270.

[4] P. Birken, K.J. Quint, S. Hartmann, and A. Meister. On coupling schemes for heat transfer in FSI applications. In S. Hartmann, A. Meister, M. Schäfer, and S. Turek, editors, *Fluid-Structure-Interaction. Theory, Numerics and Applications*, pages 21 – 30. Kassel University Press, 2009.

[5] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical solution of initial-value problems in differential-algebraic equations*, volume 14 of *Classics in Applied Mathematics.* SIAM, Philadelphia, 1996.

[6] J. W. Butcher. On Runge–Kutta processes of high order. *J. Austral. Math. Soc.*, 4:179–194, 1964.

[7] J.C. Butcher. On the implementation of implicit Runge-Kutta methods. *BIT*, 16:237–240, 1976. doi:10.1007/BF01932265.

[8] J.C. Butcher. *The numerical analysis of ordinary differential equations. Runge-Kutta and general linear methods.* John Wiley & Sons, Chichester, 1987.

[9] A. J. Chorin. Numerical solution for the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[10] J.-B. Colliat, M. Hautefeuille, and A. Ibrahimbegović. *Meso- to Macro-scale Probability Aspects for Size Effects and Heterogenous Materials Failure*, pages 253–275. ISTE, 2010. Available from: http://dx.doi.org/10.1002/9780470611364.ch14, doi:10.1002/9780470611364.ch14.

[11] T. A. Davis. Algorithm 832: Umfpack, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):166–199, 2004.

[12] T. A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, 2004.

[13] B.L. Ehle. High order A-stable methods for the numerical solution of systems of D.E.'s. *BIT*, 8:276–278, 1968. doi:10.1007/BF01933437.

[14] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes equations*. Springer-Verlag, Berlin-Heidelberg-New York, 1986.

[15] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method*. Wiley, Chichester, 2000.

[16] E. Hairer and G. Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1996.

[17] E. Hairer and G. Wanner. Stiff differential equations solved by Radau methods. *J. Comput. Appl. Math.*, 111(1-2):93–111, 1999. doi:10.1016/S0377-0427(99)00134-X.

[18] V. John. Reference values for drag and lift of a two–dimensional time dependent flow around a cylinder. *Int. J. Numer. Meth. Fluids*, 44:777–788, 2004.

[19] V. John and G. Matthies. MooNMD - a program package based on mapped finite element methods. *Comput. Visual. Sci.*, 6:163 – 170, 2004.

[20] V. John and J. Rang. Adaptive time step control for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 199:514–524, 2010.

[21] M. Krosche and M. Hautefeuille. Simulation and solution of stochastic systems with a component-based software design. *PAMM*, 7(1):2140001–2140002, 2007. Available from: http://dx.doi.org/10.1002/pamm.200700067, doi:10.1002/pamm.200700067.

[22] M. Krosche, R. Niekamp, and H. G. Matthies. PLATON: A problem solving environment for computational steering of evolutionary optimisation on the grid. In G. Bugeda, J.A.-Désidéri, J. Periaux, M. Schoenauer, and G. Winter, editors, *EUROGEN*, 2003.

[23] J. Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems*, volume 16 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2001.

[24] D. Markovic, R. Niekamp, A. Ibrahimbegović, H. G. Matthies, and R. L. Taylor. Multi-scale modeling of heterogeneous structures with inelastic constitutive behaviour: Part I - physical and mathematical aspects. *Engineering Computations: Int J for Computer-Aided Engineering*, 22(5-6):664–683, 2005. Available from: http://www.ingentaconnect.com/content/mcb/182/2005/00000022/F0020005/art00008, doi:10.1108/02644400510603050.

[25] H. G. Matthies, R. Niekamp, and J. Steindorf. Algorithms for strong coupling procedures. *Computer Methods in Applied Mechanics and Engineering*, 195(17 - 18):2028 – 2049, 2006. Available from: http://www.sciencedirect.com/science/article/pii/S0045782505001969, doi:10.1016/j.cma.2004.11.032.

[26] R. Niekamp, D. Markovic, A. Ibrahimbegović, H. G. Matthies, and R. L. Taylor. Multi-scale modelling of heterogeneous structures with inelastic constitutive behaviour: Part II - software coupling implementation aspects. *Engineering Computations: Int J for Computer-Aided Engineering*, 26(1-2):6–28, 2009. Available from: http://www.ingentaconnect.com/content/mcb/182/2009/00000026/F0020001/art00001, doi:10.1108/02644400910924780.

[27] J. Rang. A New Stiffly Accurate Rosenbrock-Wanner Method for Solving the Incompressible Navier-Stokes Equations. In R. Ansorge, H. Bijl, A. Meister, and T. Sonar, editors, *Recent Developments in the Numerics of Nonlinear Hyperbolic Conservation Laws*, volume 120, pages 301–315. Springer Verlag, Heidelberg, Berlin, 2012.

[28] J. Rang. Improved traditional Rosenbrock–Wanner methods for stiff ODEs and DAEs. Informatik-Bericht 2013-05, TU Braunschweig,

Braunschweig, 2013. Available from: http://www.digibib.tu-bs.de/?docid=00055262.

[29] J. Rang. An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4. Informatik-Bericht 2014-06, TU Braunschweig, Braunschweig, 2014. Available from: http://www.digibib.tu-bs.de/?docid=00056090.

[30] J. Rang. An analysis of the Prothero-Robinson example for constructing new DIRK and ROW methods. *Journal of Computational and Applied Mathematics*, 262:105–114, 2014. doi:http://dx.doi.org/10.1016/j.cam.2013.09.062.

[31] J. Rang. Apdative timestep control for fully implicit Runge–Kutta methods of higher order. Informatik-Bericht 2014-03, TU Braunschweig, Braunschweig, 2014. Available from: http://www.digibib.tu-bs.de/?docid=00055783.

[32] J. Rang, M. Krosche, R. Niekamp, and H. G. Matthies. Solving fluid-structure interaction problems using strong coupling algorithms with the component template library (ctl). *PAMM*, 8(1):10987–10988, 2008. Available from: http://dx.doi.org/10.1002/pamm.200810987, doi:10.1002/pamm.200810987.

[33] M. Schäfer and S. Turek. The benchmark problem "Flow around a cylinder". In E.H. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, volume 52 of *Notes on Numerical Fluid Mechanics*, pages 547–566. Vieweg, 1996.

[34] T. Srisupattarawanit, R. Niekamp, and H. G. Matthies. Simulation of nonlinear random finite depth waves coupled with an elastic structure. *Computer Methods in Applied Mechanics and Engineering*, 195(23 - 24):3072 – 3086, 2006. Incompressible CFD. Available from: http://www.sciencedirect.com/science/article/pii/S0045782505002392, doi:10.1016/j.cma.2005.02.027.

[35] K. Strehmel and R. Weiner. *Linear-implizite Runge–Kutta-Methoden und ihre Anwendung*, volume 127 of *Teubner-Texte zur Mathematik*. Teubner, Stuttgart, 1992.

[36] K. Strehmel and R. Weiner. *Numerik gewöhnlicher Differentialgleichungen*. Teubner, Stuttgart, 1995.

[37] R. Williams, K. Burrage, I. Cameron, and M. Kerr. A four-stage index 2 diagonally implicit Runge-Kutta method. *Appl. Numer. Math.*, 40(3):415–432, 2002. doi:10.1016/S0168-9274(01)00090-3.

Technische Universität Braunschweig
Informatik-Berichte ab Nr. 2011-10

2011-10	B. V. Rosić, A. Kučerová, J. Sýkora, A. Litvinenko, O. Pajonk and H. G. Matthies	Parameter Identification in a Probabilistic Setting
2011-11	M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies and E. Zander	Efficient Analysis of High Dimensional Data in Tensor Formats
2011-12	S. Oster	A Semantic Preserving Feature Model to CSP Transformation
2012-01	O. Pajonk, B. V. Rosić and H. G. Matthies	Deterministic Linear Bayesian Updating of State and Model Parameters for a Chaotic Model
2012-02	B. V. Rosić and H. G. Matthies	Stochasticc Plasticity - A Variational Inequality Formulation and Functional Approximation Approach I: The Linear Case
2012-03	J. Rang	An analysis of the Prothero–Robinson example for constructing new DIRK and ROW methods
2012-04	S. Kolatzki, M. Hagner, U. Goltz and A. Rausch	A Formal Definition for the Description of Distributed Concurrent Components - Extended Version
2012-05	M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies and P. Wähnert	Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats

2012-06	S. Mennike	A Petri Net Semantics for the Join-Calculus
2012-07	S. Lity, R. Lachmann, M. Lochau, I. Schaefer	Delta-oriented Software Product Line Test Models - The Body Comfort System Case Study
2013-01	M. Lochau, S. Mennicke, J. Schroeter und T. Winkelmann	Extended Version of 'Automated Verification of Feature Model Configuration Processes based on Workflow Petri Nets'
2013-02	S. Lity, M. Lochau, U. Goltz	A Formal Operational Semantics of Sequential Function Tables for Model-based SPL Conformance Testing
2013-03	L. Giraldi, A. Litvinenko, D. Liu, H. G. Matthies, A. Nouy	To be or not to be intrusive? The solution of parametric and stochastic equations – the "plain vanilla" Galerkin case
2013-04	A. Litvinenko, H. G. Matthies	Inverse problems and uncertainty quantification
2013-05	J. Rang	Improved traditional Rosenbrock–Wanner methods for stiff ODEs and DAEs
2013-06	J. Koslowski	Deterministic single-state 2PDAs are Turing-complete
2014-01	B. Rosić, J. Diekmann	Stochastic Description of Aircraft Simulation Models and Numerical Approaches
2014-02	M. Krosche, W. Heinze	A Robustness Analysis of a Preliminary Design of a CESTOL Aircraft
2014-03	J. Rang	Apdative timestep control for fully implicit Runge–Kutta methods of higher order

2014-04	S. Mennicke, J.-W. Schicke-Uffmann, U. Goltz	Free-Choice Petri Nets and Step Branching Time
2014-05	A. Martens, C. Borchert, T. O. Geissler, O. Spinzcyk, D. Lohmann, R. Kapitza	Exploiting determinism for efficient protection against arbitrary state corruptions
2014-06	J. Rang	An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4
2014-07	J. Rang, R. Niekamp	A component framework for the parallel solution of the incompressible Navier-Stokes equations with Radau-IIA methods